

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Martin Bubnič

Sistem za potrjevanje računov

DIPLOMSKO DELO
VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2013

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Martin Bubnič, z vpisno številko 63090190, sem avtor diplomskega dela z naslovom:

Evidenca faktur

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Rok Rupnika
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, 23. oktober 2013

Podpis avtorja:

Kazalo

POVZETEK	1
EXTRACT.....	2
1 UVOD.....	3
2 RAZLAGA DELOVANJA	4
2.1 Splošno	4
2.2 Diagrami	5
2.2.1 Diagram poteka.....	5
2.2.2 Diagram primera uporabe.....	6
2.3 Podatkovni model.....	7
2.3.1 Splošno	7
2.3.2 Podrobnosti.....	7
2.3.2.1 Podatkovni model.....	7
2.3.2.2 Logični model podatkovnega modela.....	8
2.3.2.3 Fizični model podatkovnega modela	9
2.4 Spletni del aplikacije	17
2.4.1 Splošno	17
2.4.2 Varnost	17
2.4.3 Tehnologija XSLT.....	18
2.4.4 Podrobnosti.....	19
2.4.4.1 Prijava v sistem.....	19
2.4.4.2 Statistika	20
2.4.4.3 Vnos fakture	21
2.4.4.4 Pregled faktur	24
2.5 Mobilna aplikacija	27
2.5.1 Splošno	27
2.5.2 Opreacijski sistem Android	28
2.5.3 Mobilna aplikacija evidenca faktur	28
2.5.3.1 Povezava na podatkovno bazo	28
2.5.3.2 Delovanje mobilne aplikacije	31
3 MOŽNOSTI RAZŠIRITEV SISTEMA	35
4 ZAKLJUČEK	36
LITERATURA IN VIRI.....	37

Kazalo slik

Slika 1: Model - vpogled - kontrola	4
Slika 2: Diagram poteka	5
Slika 3: Diagram primera uporabe.....	6
Slika 4: Logični model podatkovnega modela	8
Slika 5: XSL transformacija	18
Slika 6: Prijava v sistem	19
Slika 7: Pregled statistike	20
Slika 8: Vnos fakture	21
Slika 9: Dodajanje storitev	23
Slika 10: Pregled faktur	24
Slika 11: Faktura v formatu PDF.....	26
Slika 12: Prikaz delovanja spletnega servisa	30
Slika 13: Prikaz seznama v mobilni aplikaciji.....	32
Slika 14: Prikaz podrobnosti ter zaključevanje fakture	33

Kazalo tabel

Tabela 1: Vodenje faktur	9
Tabela 2: Vpogled fakture	10
Tabela 3: Davek na dodano vrednost	10
Tabela 4: Enote	11
Tabela 5: Faktura – storitev	11
Tabela 6: Faktura	12
Tabela 7: Nadzor pregleda.....	13
Tabela 8: Pravice	13
Tabela 9: Storitve	14
Tabela 10: Stranke	14
Tabela 11: Uporabniki	15
Tabela 12: Šifrant enot	15
Tabela 13: Šifrant faktur.....	16

POVZETEK

Diplomsko delo predstavlja programsko opremo za podporo evidentiranju faktur. Programska oprema je namenjena elektronskemu evidentiranju izhodnih faktur v podjetju ali organizaciji in predstavlja alternativo dosednji evidenci v papirnati obliki. Sestavljajo jo podatkovni in spletni del ter mobilna aplikacija za potrjevanje faktur. Načrtovana je s pomočjo orodja Enterprise architect, izdelana pa s pomočjo orodij Microsoft in Eclipse. Podatkovni del je implementiran z orodjem Microsoft SQL Server Managment Studio 2008 z dodatkom SSMS orodij. Spletni grafični vmesnik in logika sta implementirana v Microsoft Visual studio 2010 z dodatkom knjižnice Winnovative 6.0 na operacijskem sistemu Microsoft Windows 7. Mobilna aplikacija je razvita s pomočjo orodja Eclipse z dodatkom ksaop2. Aplikacija je narejena za platformo Android.

Ključne besede: Evidenca, faktura, Android, podjetje, aplikacija, spletni servis, grafični vmesnik, potrjevanje

EXTRACT

Thesis presents the invoice recording support software. This software is designed for electronic recording of outgoing invoices in an enterprise or organization as an alternative to the current records in paper form. It consists of three parts - data part, online part and mobile application for the invoice validation. It is designed with Enterprise architect and implemented by using Microsoft Eclipse. The data part is implemented with Microsoft SQL Server Managment Studio 2008 with the addition of SSMS. Web-based graphical interface and logic are implemented in Microsoft Visual Studio 2010 with the addition of the Winnovative 6.0 on Microsoft Windows 7. Mobile application was developed using the Eclipse by adding ksaop2. The application is designed for the Android platform.

Keywords: records, invoices, Android, enterprise, application, web service, Graphical User Interface, validation

1 UVOD

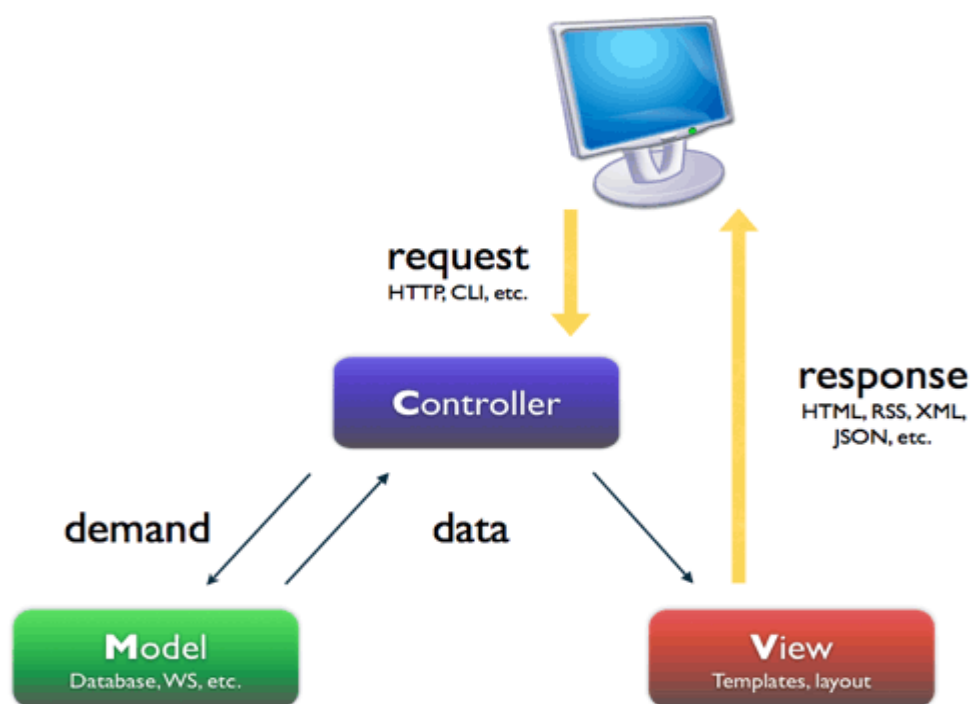
V informacijski dobi, v kateri se nahajamo, se človeške procese poskuša informatizirati, ter tako olajšati. Informatizacija nam daje preprostejši način izvajanja človeških procesov, kot so pošiljanje sporočil, branje novic, upravljanje s transakcijami, itd. Informacijski sistem je prilagojen določeni fizični osebi z dejavnostjo, ki je hotela informatizirati svojo evidenco faktur. Diplomsko delo opisuje informacijski sistem, imenovan Evidenca faktur, kateri vsebuje podatkovni del v obliki podatkovne baze in uporabniški vmesnik kot spletni del informacijskega sistema za komunikacijo človek - računalnik ter mobilno aplikacijo, ki omogoča pregled ter potrjevanje nezaključenih faktur. Podatkovni del je namenjen varnemu in urejenemu hranjenju podatkov ter enostavnemu manipuliranju s podatki. Uporabniški vmesnik je spletni vmesnik, ki človeku omogoča manipuliranje s podatki ter dodajanje logično povezanih podatkov. Omogoča nam dodajanje novih strank, dodajanje novih storitev, kreiranje novih faktur, izvoz faktur v PDF dokument ter pregled nad fakturami. Mobilna aplikacija omogoča pregled in potrjevanje faktur. Informacijski sistem je prilagojen določeni fizični osebi z dejavnostjo, ki je hotela informatizirati svojo evidenco faktur in je zasnovan z možnostjo nadaljnega razvoja za več fizičnih ali pravnih oseb.

2 RAZLAGA DELOVANJA

2.1 Splošno

Evidenca faktur deluje na principu »Model – vpogled – kontrola (MVC)«. Sestavni deli MVC arhitekture so:

- Model: predstavlja podatkovni model oziroma podatkovno bazo oziroma podatke.
- Vpogled (View): predstavlja grafični vmesnik, ki je namenjen komunikaciji na relaciji človek → računalnik in računalnik → človek.
- Kontrola (Controller): predstavlja logiko sistema. Z modelom komunicira tako, da manipulira s podatki ali poizveduje po njih, ter z vpogledom za prikazovanje določenih informacij uporabniku (človeku).



Slika 1: Model - vpogled - kontrola

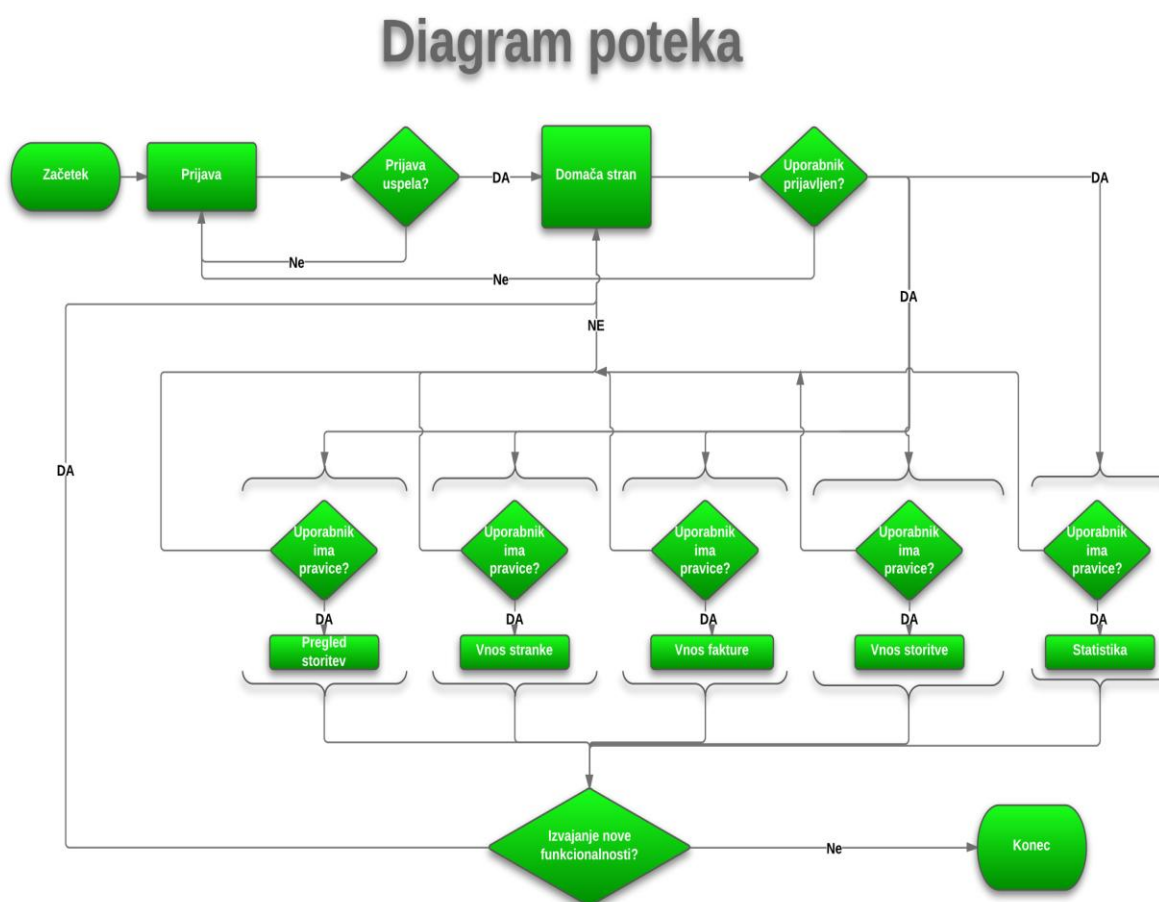
Informacijski sistem »Evidenca faktur« vsebuje:

- Podatkovni del: podatkovni del nam omogoča varno in urejeno hranjenje podatkov ter manipuliranje z njimi. Podatkovni model vsebuje deset tabel, en vpogled ter tri metode. Več o podatkovnem modelu v naslednjem poglavju.
- Uporabniški vmesnik: Uporabniški vmesnik vsebuje prijavo, vnos novih strank, vnos novih storitev, kreiranje faktur ter pregled faktur. Uporabnik mora imeti za vsako stran pravice, tako lahko vsakemu uporabniku posebej določamo katere aktivnosti sme izvajati. Za večji nadzor pred zlorabami spletne aplikacije se vsak pogled nadzoruje z beleženjem, prijava v sam sistem pa je kodirana po določenem kriptografskem postopku MD5. Več o celotnem spletnem delu v naslednjem poglavju.

- Mobilno aplikacijo: Mobilna aplikacija je zasnovana na platformi Android. Deluje na Android operacijskih sistemih od verzije 2.3.3 dalje. Mobilna aplikacija omogoča pregledovanje nepotrjenih faktur, prikaz fakturnih podrobnosti ter potrjevanje faktur. Več o delovanju mobilne aplikacije v posebnem poglavju.

2.2 Diagrami

2.2.1 Diagram poteka

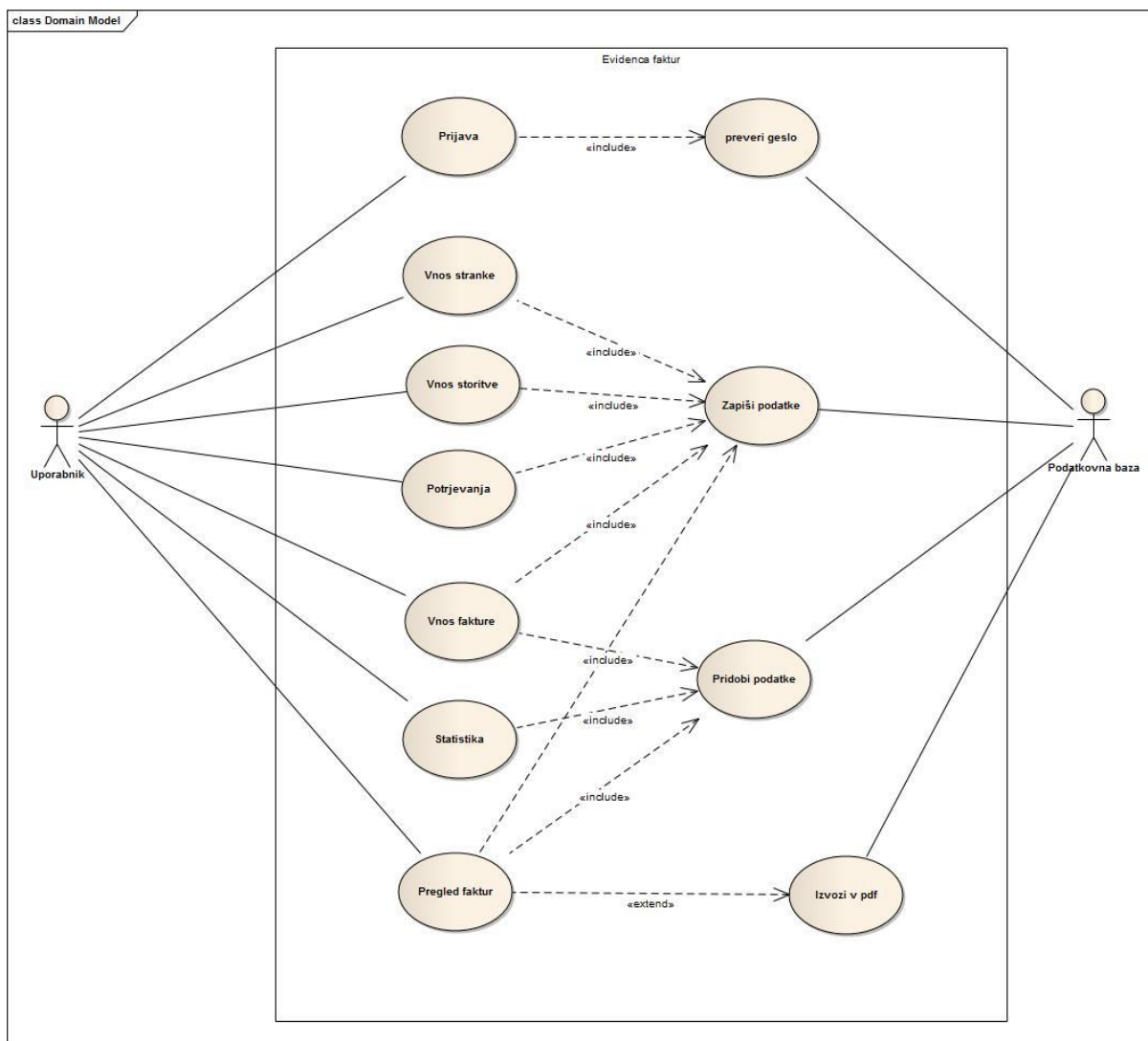


Slika 2: Diagram poteka

Diagram poteka (angl. *flow chart*) je diagram za prikaz možnih poti podatkov skozi sistem oz. program ali eden izmed načinov zapisa algoritma. Uporablja se v računalništvu, pa tudi v pravi, medicini, matematiki in še v mnogih drugih vedah. Diagrame poteka se večina riše v UML (Diagrami aktivnosti), ki skuša postati standardiziran opisni jezik. Za opisovanje poslovnih procesov se je v zadnjem času uveljavila tudi grafična notacija BPMN.

Diagram poteka programa ali kar diagram poteka uporabljamo za opis poteka operacij določenega računalniškega programa. Diagram prikazuje natančno zaporedje operacij, ki jih program pri obdelavi podatkov izvede. Različni grafični simboli predstavljajo vnos in izpis podatkov, odločitve, razvejitve in podprograme.

2.2.2 Diagram primera uporabe



Slika 3: Diagram primera uporabe

Diagram primera uporabe je najenostavnejši prikaz interakcij uporabnika s sistemom in prikazuje specifikacije primera uporabe. Diagram primera uporabe lahko prikazuje različne vrste uporabnikov sistema ter različne načine komunikacije s sistemom. Diagrami primera uporabe prikazujejo kakšne funkcionalnosti ima sistem, vendar ne razkriva podrobnosti, kako to počne. Primerni so za načrtovanje kakršnih koli računalniških sistemov, za predstavitev sistema širši skupini uporabnikov in tudi stranki.

2.3 Podatkovni model

2.3.1 Splošno

Spodaj predstavljeni podatkovni model je zasnovan v orodju Microsoft SQL Server Managment Studio 2008 z dodatkom SSMS tools. Kreiran je v jeziku T-SQL, po principu »DDL« in »DML«. DDL (Data definition language) predstavlja sintakso kreiranja raznoraznih objektov, kot so tabele, vpogledi, procedure, prožilci, itd. DML (Data manipulation language) predstavlja sintakso manipuliranja s podatki, npr.: selekcija, projekcija, popravljanje, brisanje in vnašanje.

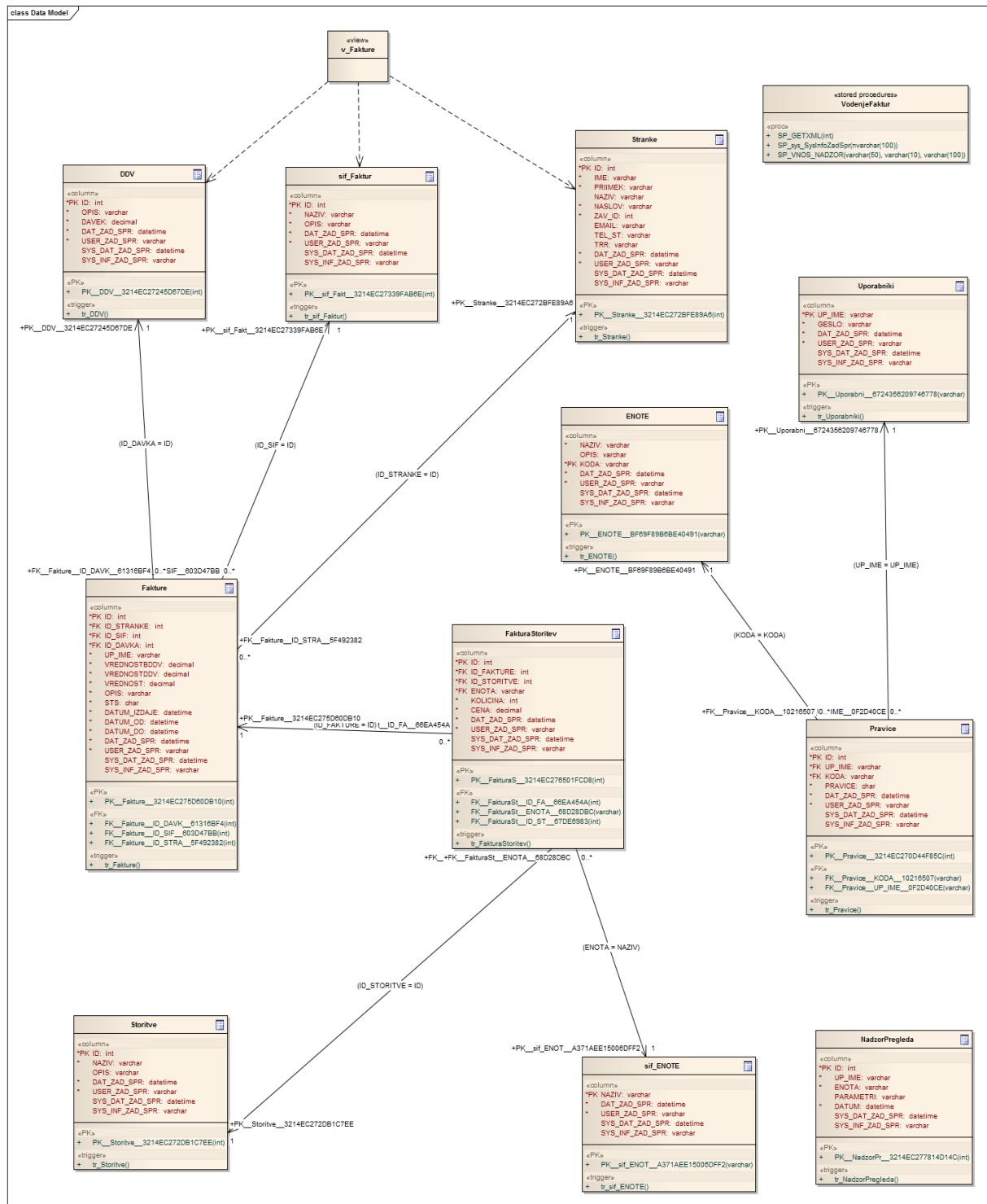
Narejen je po merilih točno določene fizične osebe z dejavnostjo, ki je predpisala katere informacije potrebuje o določeni stranki, storitvi in fakturi. Podatkovni model informacijskega sistema sestavljajo tabele, procedure in vpogled. Vsebuje deset tabel, tri procedure in en vpogled. Tabele so povezane v logično strukturo preko primarnih in tujih ključev, v katerih se hranijo vsi potrebni podatki za evidentiranje, uspešno in učinkovito iskanje, pregledovanje, urejanje in dodajanje podatkov o fakturah. Nad vsako tabelo bedi prožilec, ki se ob vsakem vnosu ali popravku kateregakoli podatka sproži, s tem se evidentira vsaka sprememba v podatkih ter omogoča učinkovit nadzor nad podatki, tako v primeru zlorab kot v primeru napak. Do podatkovnega modela je mogoče dostopati preko spletne aplikacije, v kateri je logika za dostopanje do podatkov v podatkovni bazi, mogoče pa je dostopati tudi preko orodja Microsoft SQL Server Managment Studio 2008 s pomočjo T-SQL jezika s konceptom DML.

2.3.2 Podrobnosti

2.3.2.1 Podatkovni model

<i>Ime:</i>	<i>VodenjeFaktur</i>
<i>Podatkovna baza:</i>	<i>SQL Server 2008</i>
<i>Podrobnosti:</i>	<i>Kreirano dne 3.3.2013.</i>
<i>Avtor:</i>	<i>Martin Bubnič</i>
<i>Zadnja sprememba:</i>	<i>12.4.2013</i>
<i>Verzija:</i>	<i>1.0</i>

2.2.3.2 Logični model podatkovnega modela



Slika 4: Logični model podatkovnega modela

2.2.3.3 Fizični model podatkovnega modela

VodenjeFaktur

Podatkovna baza: SQL Server 2008

Tip: »Metode«

Ime	Tip	Vhodni parametri
SP_GETXML	Javno	@ID
SP_sys_SysInfoZadSpr	Javno	@SYSINFO_ZAD_SPR
SP_VNOS_NADZOR	Javno	@UP_IME @ENOTA @PARAMETRI

Tabela 1: Vodenje faktur

Opis:

1. SP_GETXML: Namen metode »SP_GETXML« je vračanje podrobnosti določene fakture. Kot vhodni parameter ji podamo enolični identifikator (ID) fakture, kot izhod pa dobimo rezultat v xml obliki, v katerem so podatki o stranki, fakturi in vseh storitvah. Metoda »SP_GETXML« pred vračanjem rezultata popravi status fakture na status izdane fakture.
2. SP_sys_SysInfoZadSpr: Namen metode »SP_sys_SysInfoZadSpr« je vračanje sistemskih informacij trenutnega časa. To so: Uporabnik, sistemski uporabnik, enolični identifikator računalnika, ime računalnika, ime programa. Vsaka tabela v podatkovni bazi vsebuje polje »SYSINFO_ZAD_SPR« kar pomeni sistemske informacije zadnje spremembe. To polje se polni preko prožilca nad vsako tabelo, po vnosu ali spreminjanju. Vrednost polja določa metoda.
3. SP_VNOS_NADZOR: Zaradi boljšega nadzora nad uporabniki ter morebitnimi zlorabami, se v spletni aplikaciji »Evidenca faktur« vsako pregledovanje beleži. Za to poskrbi metoda »SP_VNOS_NADZOR«. Kot vhodne parametre ji podamo uporabnika, enoto na kateri se nahaja (stran v spletni aplikaciji) in vse navedene parametre. Metoda nato v tabelo »Nadzor« zabeleži kdo, kdaj in s kakšnimi parametri je opravljal pregled.

v_Fakture

Podatkovna baza: SQL Server 2008

Tip: »Vpogled«

Relacije

Columns	Association	Notes
	Stranke.	
	sif_Faktur.	
	DDV.	

Tabela 2: Vpogled fakture

Opis:

1. Vpogled v_Fakture povezuje več tabel, ter tako črpa potrebne podatke za prikaz faktur. Povezuje tabele »Stranka«, »sif_Faktur«, »DDV«, ter Faktura. Vpogled hrani točno določene podatke, katere prikazujemo na pregledu. S tem se ob pregledovanju faktur izognemo preformančnim težavam s povezovanjem omenjenih tabel.

DDV

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	ID	int	Da	Ne	0	0	0
Ne	OPIS	varchar	Da	Ne	50	0	0
Ne	DAVEK	decimal	Da	Ne	0	18	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 3: Davek na dodano vrednost

Opis:

1. Tabela »DDV« je šifrantska tabela možnih davkov. Namenjena je hranjenju šifranta davkov na primer: 8,5%, 20%, ..

Enote

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Ne	NAZIV	varchar	Da	Ne	100	0	0
Ne	OPIS	varchar	Ne	Ne	100	0	0
Da	KODA	varchar	Da	Ne	10	0	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 4: Enote

Opis:

1. Spletna aplikacija »Evidenca faktur« vsebuje več spletnih strani. Vsaka stran je označena z določeno enoto npr. (ASPX01). Vsakemu uporabniku lahko tako določimo za katere enote ima pravice.

Faktura - Storitve

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	ID	int	Da	Ne	0	0	0
Ne	ID_FAKTURE	int	Da	Ne	0	0	0
Ne	ID_STORITVE	int	Da	Ne	0	0	0
Ne	ENOTA	varchar	Da	Ne	100	0	0
Ne	KOLICINA	int	Da	Ne	0	0	0
Ne	CENA	decimal	Da	Ne	0	9	2
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 5: Faktura – Storitve

Opis:

1. Tabela »Faktura – Storitve« je vmesna pomožna tabela, ki povezuje tabele »Faktura«, »Storitve« ter »sif_Enote«. Povezuje jih preko tujih ključev (»ID_FAKTURE«, »ID_STORITVE«, »ENOTA«). Ker lahko v določeni fakturi naštejemo več storitev ter za vsako storitev določimo enoto in vnesemo ceno, potrebujemo to pomožno tabelo.

Faktura

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	ID	int	Da	Ne	0	0	0
Ne	ID_STRANKE	int	Da	Ne	0	0	0
Ne	ID_SIF	int	Da	Ne	0	0	0
Ne	ID_DAVKA	int	Da	Ne	0	0	0
Ne	UP_IME	varchar	Da	Ne	50	0	0
Ne	VREDNOSTBDDV	decimal	Da	Ne	0	9	2
Ne	VREDNOSTDDV	decimal	Da	Ne	0	9	2
Ne	VREDNOST	decimal	Da	Ne	0	9	2
Ne	OPIS	varchar	Da	Ne	50	0	0
Ne	STS	char	Da	Ne	1	0	0
Ne	DATUM_IZDAJE	datetime	Da	Ne	0	0	0
Ne	DATUM_OD	datetime	Da	Ne	0	0	0
Ne	DATUM_DO	datetime	Da	Ne	0	0	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 6: Faktura

Opis:

1. Točno določeno fakture lahko izdamo določeni stranki, z določeno stopnjo davka. Tabela »Faktura« hrani fakture ter potrebne podrobnosti o fakturi. Povezuje se s tremi tabelami preko tujih ključev, to so: »Stranke«, »sif_Faktur«, »DDV«. Polje »ID_STRANKE« se navezuje na stranko, kateri je izdana faktura, polje ID_SIF se navezuje na sifrantsko tabelo. Označi se vrsto fakture (ali gre za račun, predračun, dobavnico...). »ID_DAVKA« se navezuje na dddv, ter tako odmeri davek fakturi. Po polju »STS« prepoznamo ali gre za izdano fakture ali pripravljeno fakture. V tabeli »Faktura« se hrani tudi skupno ceno fakture, ceno brez dddv, vrednost dddv, datum izdaje fakture, časovni interval opravljanja storitve.

NadzorPregleda

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarno ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	ID	int	True	False	0	0	0
Ne	UP_IME	varchar	True	False	50	0	0
Ne	ENOTA	varchar	True	False	10	0	0
Ne	PARAMETRI	varchar	False	False	100	0	0
Ne	DATUM	datetime	True	False	0	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	False	False	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	False	False	200	0	0

Tabela 7: Nadzor pregleda

Opis:

1. V spletni aplikaciji »Evidenca faktur« se vsak dogodek pri pregledu beleži v posebno tabelo. Vnaša se v tabelo »NadzorPregleda«. Eno beleženje sestavlja uporabniško ime, enota strani, parametri ter datum.

Pravice

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	ID	int	Da	Ne	0	0	0
Ne	UP_IME	varchar	Da	Ne	50	0	0
Ne	KODA	varchar	Da	Ne	10	0	0
Ne	PRAVICE	char	Da	Ne	1	0	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Da	Ne	200	0	0

Tabela 8: Pravice

Opis:

1. Vsakemu uporabniku lahko določimo novo kodo (stran), do katere ima dostop. Ne omogoča nam dodeljevanja pravic celotnim skupinam uporabnikov (ADMIN, USER) ampak posamezno.

Storitve

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
True	ID	int	True	False	0	0	0
False	NAZIV	varchar	True	False	100	0	0
False	OPIS	varchar	False	False	200	0	0
False	DAT_ZAD_SPR	datetime	True	False	0	0	0
False	USER_ZAD_SPR	varchar	True	False	50	0	0
False	SYS_DAT_ZAD_SPR	datetime	False	False	0	0	0
False	SYS_INF_ZAD_SPR	varchar	False	False	200	0	0

Tabela 9: Storitve

Opis:

1. V tabeli »Storitev« se hrani nabor storitev, ki jih tržimo. Dodaja se jih v aplikaciji. Vsaka storitev je lahko dodeljena določeni fakturi.

Stranke

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarno ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	ID	int	Da	Ne	0	0	0
Ne	IME	varchar	Da	Ne	50	0	0
Ne	PRIIMEK	varchar	Da	Ne	50	0	0
Ne	NAZIV	varchar	Ne	Ne	100	0	0
Ne	NASLOV	varchar	Da	Ne	200	0	0
Ne	ZAV_ID	int	Da	Ne	0	0	0
Ne	EMAIL	varchar	Ne	Ne	100	0	0
Ne	TEL_ST	varchar	Ne	Ne	30	0	0
Ne	TRR	varchar	Ne	Ne	50	0	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 10: Stranke

Opis:

1. V tabeli »Stranke« se hrani nabor strank, ki jim želimo oziroma smo jim že izdali fakturo. Dodaja se jih v aplikaciji. Vsaki stranki lahko izdamo poljubno število faktur.

Uporabniki

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	UP_IME	varchar	Da	Ne	50	0	0
Ne	GESLO	varchar	Da	Ne	50	0	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 11: Uporabniki

Opis:

1. V tabeli »Uporabniki« se hranijo uporabniška imena uporabnikov ter njihova gesla, potrebna za prijavo v aplikacijo. Zaradi verjetnosti zlorabe je geslo kriptirano s kriptografsko funkcijo imenovano »MD5«.

sif_Enote

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	NAZIV	varchar	Da	Ne	100	0	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 12: Šifrant enot

Opis:

1. Spletna aplikacija »Evidenca faktur« omogoča izdajo fakture, na kateri so razne storitve. Storitve imajo lahko različne enote npr.: kos, ura, m2. Tabela »ENOTE« je šifrantska tabela možnih enot. Namenjena je hranjenju šifrantskih enot, ki jih izberemo pri storitvi.

sif_Faktur

Podatkovna baza: SQL Server 2008

Tip: »Tabela«

Atributi

Primarni ključ	Ime	Tip	Obvezen	Unikaten	Dolž	Omej	Skal
Da	ID	int	Da	Ne	0	0	0
Ne	NAZIV	varchar	Da	Ne	50	0	0
Ne	OPIS	varchar	Da	Ne	100	0	0
Ne	DAT_ZAD_SPR	datetime	Da	Ne	0	0	0
Ne	USER_ZAD_SPR	varchar	Da	Ne	50	0	0
Ne	SYS_DAT_ZAD_SPR	datetime	Ne	Ne	0	0	0
Ne	SYS_INF_ZAD_SPR	varchar	Ne	Ne	200	0	0

Tabela 13: Šifrant faktur

Opis:

1. Šifranstska tabela »sif_faktur« se uporablja za določitev vrste fakture in pove ali se uporablja račun, predračun ali dobavnico.

2.4 Spletni del aplikacije

2.4.1 Splošno

Spletni del aplikacije je zasnovan v orodju Microsoft visual studio, v programskih jezikih C#, html (Hiper text marktup language), xslt, javascript. Namenjen je enostavnemu manipuliranju s podatki, povezovanju le-teh v logične strukture in prikazu informacij. Načrtovan je po metodologiji hitrega razvoja (RAD). Metodologija hitrega razvoja za uskladitev zahtev z naročnikom uporablja prototipe, na katere naročnik podaja pripombe, dokler ni zadovoljen.

Ko je prototip potrjen, nastopi implementacija samega uporabniškega vmesnika.

Spletni del sestavlja deset ».aspx« strani s skupno osnovno stranjo imenovano »MasterPage«. Omogoča nam prijavo v sistem s srednjenizko stopnjo varnosti, odjavo, vnašanje podatkov o strankah in storitvah, vnos fakture, pregled in iskanje faktur, izvoz v pdf ter vpogled v statistične podatke.

2.4.2 Varnost

Pri informacijskih sistemih, ki temeljijo na dostopnosti preko svetovnega spletnega omrežja, je uporaba varnostnih mehanizmov zelo pomembna. Za prijavo v sistem poznamo ogromno mehanizmov, kot so kriptirana gesla, certifikati, dodatno kompleksno berljive kode, s katerimi se izognemo napadu sistema z barvnimi tabelami. Za preprečitev zlorabe poznamo dodeljevanje pravic uporabnikom ter beleženje glavnih funkcij. Za preprečevanje napadov na sistem lahko poskrbimo tudi v sami kodi, v kateri delamo poizvedbe na podatkovno bazo. S tem problemom se je pred nekaj leti ukvarjal tudi Microsoft.

Primer napada:

Naša koda:

var Mesto;

Mesto = »Postojna«;

*var sql = "Select * From Mesta Where Mesto = " + Mesto + "";*

Poizvedba: *Select * From Mesta where Mesto = 'Postojna'*

Napadalec:

Spremenljivki Mesto doda sledečo kodo: *;drop table Mesta – '*

Poizvedba: *Select * From Mesta Where Mesto = 'Postojna';drop table Mesta--'*

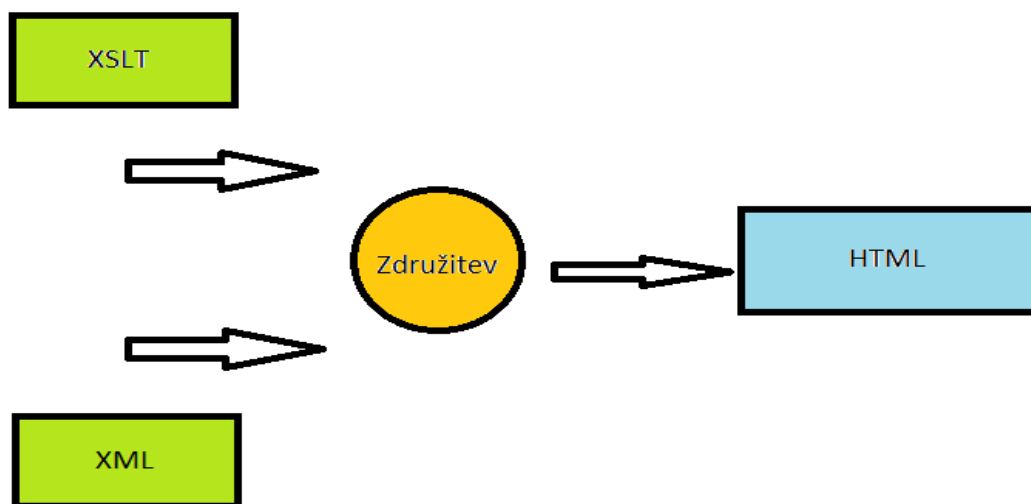
V tem primeru napadalec izbriše tabelo »Mesta« v naši podatkovni bazi. Na tak način napadalec z malo truda uniči celotno podatkovno bazo ter podatke v njej.

V informacijskem sistemu »Evidenca faktur« so uporabljene metode za preprečevanje napada pri prijavi, možnost zlorab ter tako imenovanih »SQL inject« napadov. Pri prijavi v sistem se vpisano geslo kodira s kriptografsko funkcijo MD5 ter tako poizveduje na podatkovno bazo. Tudi v podatkovni bazi so gesla shranjena kriptirana s kriptografsko funkcijo MD5. Ta način shranjevanja gesel napadalcu onemogoči pridobivanje nekriptiranih gesel tudi če pridobi podatke iz podatkovne baze. Način prijave pa ni odporen na tako imenovano »snifanje« prometa. Primer: napadalec pregleduje ves promet v omrežju, dokler se uporabnik ne prijavi. Ko se uporabnik prijavi, napadalec vidi njegov paket ter ga lahko ob zlorabi posreduje strežniku. Na ta način se lahko prijavi kot žrtev napada. Za preprečevanje takšnih napadov se uporablja metoda dodajanja »soli«. To pomeni, da strežnik pošlje uporabniku naključen niz, katerega uporabnikov sistem kriptira zraven gesla. Tako je izhodni paket za prijavo dinamičen in nikoli enak. Trenutno najmočnejša varnost avtentikacije predstavljajo certifikati.

Informacijski sistem »Evidenca faktur« za preprečevanje zlorab omogoča dodeljevanje pravic vsakemu uporabniku posebej ter beleženje pregledovanja podatkov. Tudi če bi prišlo do zlorabe, informacijski sistem v podatkovno bazo beleži vpoglede. Tako lahko sistemski administrator preverja kdo je kaj počel.

V sami kodi sistema se preprečuje napade z dodajanjem vrednosti(sql inject) na ta način da se pri poizvedovanju na podatkovno bazo uporablja metode katere so shranjene v podatkovni bazi ter parametre.

2.4.3 Tehnologija XSLT



Slika 5: XSL transformacija

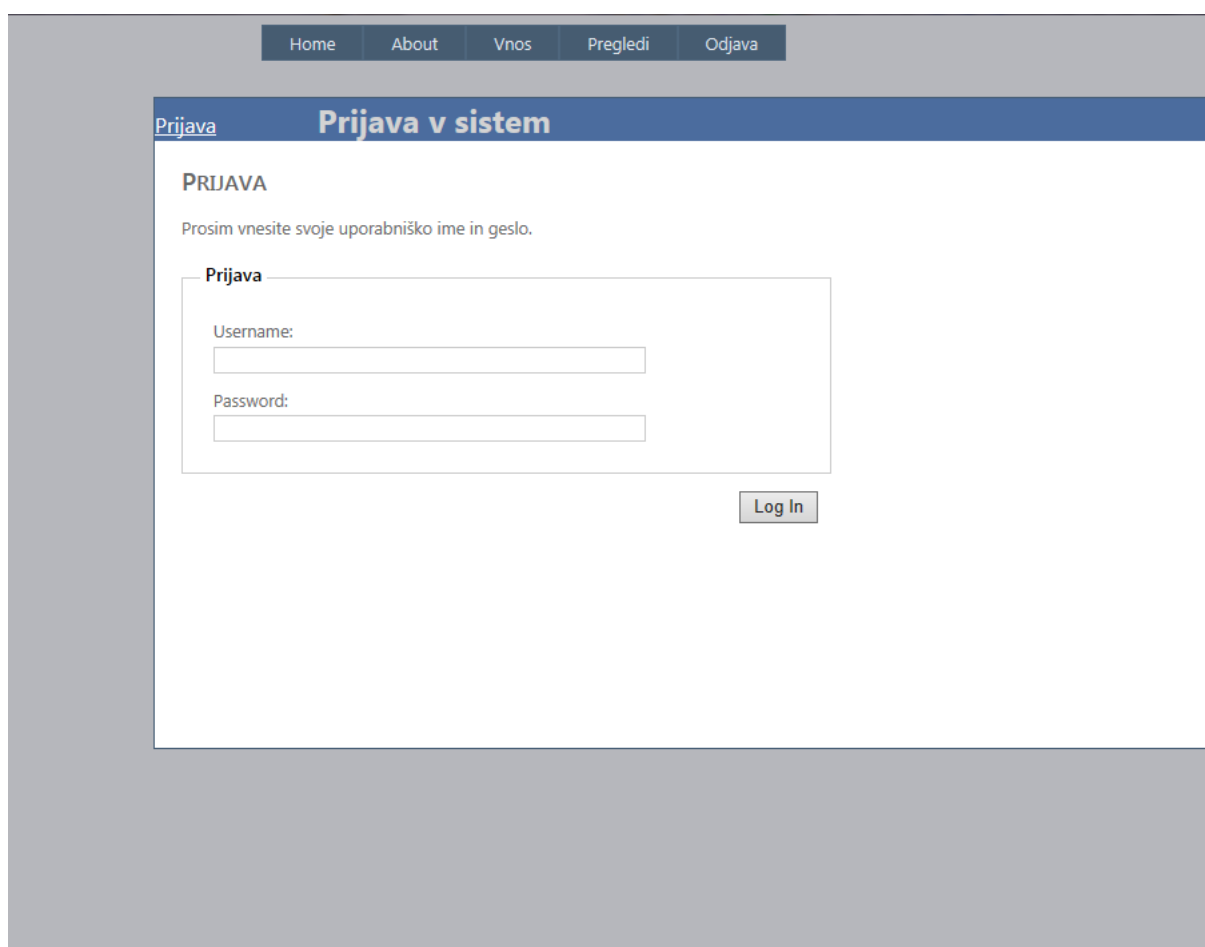
Tehnologija xslt omogoča transformacijo podatkov v html jezik. V informacijski sistem »Evidenca faktur« podatke iz xml strukture preko xslt transformira v ustrezno html strukturo, katero uporabnikov brskalnik brez težav prikaže.

2.4.4 Podrobnosti

2.4.4.1 Prijava v sistem

Prijava v sistem je najbolj pomemben varnostni mehanizem, ki mora biti zasnovan tako, da različnim uporabnikom dodeljuje različne pravice. Pri avtentikaciji v sistem se preverja uporabniško ime in geslo ter dodeljuje pravice za dostop do določenih funkcionalnosti informacijskega sistema. Geslo se kriptira s kriptografsko funkcijo MD5, kakršno se hrani tudi v podatkovni bazi.

Prikaz strani za logiranje:



The screenshot shows a web application interface for logging in. At the top, there is a navigation bar with links: Home, About, Vnos, Pregledi, and Odjava. Below this, the main content area has a header with 'Prijava' and 'Prijava v sistem'. The main section is titled 'PRIJAVA' and contains the instruction 'Prosim vnesite svoje uporabniško ime in geslo.' Below this, there is a form with two input fields: 'Username:' and 'Password:'. A 'Log In' button is located at the bottom right of the form.

Slika 6: Prijava v sistem

Pri logiranju v sistem se v sejo (spremenljivka, ki traja do zaprtje okna, do odjave ali do poteka seje) shrani uporabniško ime ter njegove pravice. Aplikacija nato uporabnika postavi na domačo stran. V primeru, da uporabnik napačno vnese uporabniško ime, aplikacija vrne opozorilo v obliki dialoga. V primeru, da uporabnik ne vnese pravilnega gesla, aplikacija vrne opozorilo v obliki izpisa »neuspešen poizkus«.

2.4.4.2 Statistika

Pregled statističnih podatkov nam omogoča pregled statističnih podatkov ter tako olajša pregled nad poslovanjem. Statistične podatke se lahko filtrira po letu, v obratnem primeru se statistično poročilo generira na podlagi vseh faktur. Izbira omogoča štiri vrste statističnih poročil:

- Statistika faktur: Statistika faktur vrne rezultat v obliki tabele. Podatki so razdeljeni po posameznem letu, prikazujejo pa skupno vrednost vseh faktur ter število vseh faktur. V statistično poročilo so zajete le fakture tipa »Račun«, ki so zaključene (plačilo je bilo izvedeno).
- Statistika strank: Statistika strank prikaže tabelarni prikaz strank za določeno leto ter število izdanih faktur, ki so ji bile izdane. Zajema fakture tipa »Račun« in so zaključene (plačilo je bilo izvedeno). Namen statistike strank je ohranjanje in vzpostavljanje kontakta z najbolj zvestimi strankami ter v obliki voščilnic, novosti, možnosti izboljšav, itd.
- Statistika storitev: Statistika storitev prikaže tabelarni prikaz storitev za posamezno leto, za katero se prikaže naziv storitve ter število naročil. Na podlagi teh podatkov lahko uporabnik naredi analizo ter investira v povečanje prodaje določene fakture. V nasprotnem primeru pridobi podatke katera storitev gre najboljše v promet.
- Statistika neplačnikov: Statistika neplačnikov prikaže vse neplačnike v danem letu. Tako lahko uporabnik vodi evidenco o neplačnikih ter lažje identificira zavezance, ki jih nato poziva k plačilu.

Prikaz statistike:

Home About Vnos Pregledi Odjava

Statistika

Izbor statistike

Izberi statistiko: Statistika storitev

Leto

Izberi leto: 2013

Najdi

Skupaj zapisov: 2

Leto	Naziv storitve	Število
2013	Montiranje radiatorja	1
2013	Montaža kotla	2

Slika 7: Pregled statistike

2.4.4.3 Vnos fakture

Vnašanje fakture je najbolj pomembna funkcionalnost informacijskega sistema »Evidenca faktur«. Na preprost, uporabniku prijazen način, omogoča vnos fakture za določeno stranko. V primeru, da stranke še ni v registru strank, jo lahko na predhodni strani dodamo. Pri vnašanju strank sistem omogoča dva načina vnašanja stranke. Če gre za pravno osebo, se vnese tudi naziv stranke, drugače se to polje izpusti. O strankah sistem hrani osebne podatke: ime, priimek, naziv, davčno številko, transakcijski račun ter naslov. Kontaktni podatki: elektronski naslov ter telefonska številka v primeru, da nam stranka zaupa podatke.

V primeru, da bi bila tržena nova storitev, mora biti najprej vnešena v register storitev. To sistem omogoča na novi strani za vnos storitev za katere hrani naslednje podatke: naziv storitve ter opis. Ko sta dva od predpogojev urejena, sistem omogoča vnašanje faktur.

Prikaz vnosa fakture:

Vnos fakture

VNESITE FAKTURO

Izbor vrste fakture

Izberite vrsto fakture Račun

Izbor stranke

Izberite stranko Test d.o.o.

Storitve

Skupaj zapisov: 2

Naziv storitve	Cena na enoto	Enota	Količina	
Montaža Bojlerja	10,00	URA	46	Odstrani[-]
Montiranje radiatorja	35,00	KOS	17	Odstrani[-]

Dodaj storitev

Izbor davka

Izberite davek 20 Izračun

Cene

Cena brez ddv 1055
Vrednost ddv 211
Cena (skupaj) 1266

Rok dela

Datum od:*
Datum do:*

Polja označena z * so obvezna.

Shrani

Slika 8: Vnos fakture

Vnos fakture poteka v več korakih. Najprej se iz podatkovne baze napolnijo šifranti za vrsto fakture, stranko ter davek. V spustni seznam za izbiro vrste fakture se najprej vnesejo šifranti vrste faktur. Ker sistem trenutno operira samo z računi in ne z dobavnicami se v spustni seznam vnese račun ter predračun. V spustni seznam za izbiro stranke se vnesejo vse stranke (njihovi nazivi oziroma imena in priimki) katere sistem hrani v registru strank. Po naročilu stranke, katera operira samo z davčno stopnjo 20 odstotkov se v spustni meni za izbiro davka napolni vrednost iz šifrantske tabele davkov. Trenutno se v šifrantski tabeli hrani samo vrednost 20 odstotkov, z datumskim rokom 1.7.2013 pa se davčna stopnja povečuje za dva odstotka na 22 odstotkov. V primeru, da bo stranka želela operirati tudi z nižjim davkom se bo šifrantska tabela davkov dopolnila.

Po izboru vrste fakture in določene stranke sistem zahteva vnos storitev, ki jih tržimo z izdano fakto. Proces dodajanja storitev je opisan v naslednjem koraku. Po dodajanju storitve se nam na zaslonu prikažejo vse storitve s podatki: cena na enoto, enota, količina ter možnost odstranitve storitve. Ko je proces dodajanja storitev zaključen nastopi proces izbora davka ter izračun. V spustnem seznamu sistem omogoča izbiro stopnje davka, v našem primeru trenutno samo 20 odstotkov. Po kliku na gumb izračun sistem izračuna cene fakture in prikaže na zaslon. Izračun poteka v treh korakih. Najprej izračuna ceno za posamezno storitev (produkt količine in cena na enoto). Ko za vse storitve izračuna ceno izračuna seštevek vseh storitev skupaj. Tako dobimo ceno fakture brez davka na dodano vrednost. Izračun vrednosti davka na dodano vrednost sistem izvede s procentnim računom za točno določeno izbrano davčno stopnjo. Tako dobimo vrednost davka na dodano vrednost. Skupna vrednost fakture znaša sumarna vrednost davka na dodano vrednost in skupna vrednost storitev. Sistem izračuna vse tri podatke ter jih prikaže v vrstnem redu: cena brez davka na dodano vrednost, vrednost davka na dodano vrednost ter skupna vrednost fakture. Skladno z zakonodajo Republike Slovenije in z obrazcem mora vsak samostojni podjetnik na izdani fakturi navesti datumski rok izvedbe dela oziroma storitve, zato obrazec v sistemu zahteva tudi vnos delovnega obdobja.

Po končanem izpolnjevanju obrazca za vnos fakture s klikom na gumb shrani sistem v podatkovno bazo podatke v naslednjem vrstnem redu:

- Shranjevanje podatkov o fakturi: Shranjevanje podatkov o fakturi poteka preko metode, ki je shranjena v podatkovni bazi. Vrednosti so parametrizirane, tako metoda kot vhod dobi parametre s podatki o fakturi ter jih nato zapiše v tabelo faktur. Podatki o fakturi, ki se vnašajo v podatkovno bazo so:
 - Šifra stranke: Vsako fakto se lahko izda samo točno določeni stranki.
 - Vrsta fakture: Lahko se izda račun ali predračun.
 - Stopnja davka: Trenutno na osnovi 20 odstotkov, z zakonskim datumom 1.7.2013 na osnovi 22 odstotkov. Po zahtevi stranke tudi nižji davek 8,5 odstotkov oziroma z novim zakonskim datumom 9,5 odstotka.
 - Cena brez ddv: Izračun opisam v zgornjem poglevju.
 - Vrednost ddv: Izračun opisam v zgornjem poglevju.
 - Skupna vrednost: Izračun opisam v zgornjem poglevju.
 - Datumsko obdobje dela: datum začetka ter datum konca dela.
- Shranjevanje podatkov o storitvah: Shranjevanje podatkov o storitvah ravno tako poteka preko metode v podatkovni bazi, katera kot vhodni parameter dobi enolični

identifikator vnešene fakture, enolični identifikator storitve, količino, enoto ter ceno na enoto. Metoda nato podatke shrani v tabelo imenovano »FakturaStoritev«, v kateri se poveže storitev s podatki o količini in ceni z fakture. Ker lahko v vsaki fakturi izdamo poljubno mnogo storitev in ker lahko storitev izdamo v poljubno mnogo fakturah, vsebuje sistem pomožno tabelo, v kateri je rešen zgoraj prikazani problem. Shranjevanje storitev poteka v zanki, tako da sistem zajame vse dodane storitve.

- Shranjevanje fakture zaradi možnosti napake poteka v transakciji. To z enostavnimi besedami pomeni, da se mora izvesti celoten korak. Torej če uspešno vnesemo fakture ter podatke o fakturi, nato nastopi napaka pri shranjevanju storitev za določeno fakture, se transakcija razveljavi, s tem se razveljavi tudi vnos fakture ter podatkov o njej. V primeru, da se napaka ne pojavi se transakcija potrdi ter tako zapiše celotni korak v podatkovno bazo. S transakcijo se izognemo napakam v podatkih, zagotavlja se celovitost podatkov in odpravlja podvajanje.

Prikaz dodajanja storitev:

Vnos fakture

VNESITE FAKTURO

Izbor vrste fakture
 Izberite vrsto fakture Račun

Izbor stranke
 Izberite stranko Test d.o.o.

Storitve
 Skupaj zapisov: 1

Naziv storitve	Cena na enoto	Enota	Količina	
Montaža kotla	10,00	URA	30	Odstrani

Dodajanje storitve

Enota URA
 Cena na enoto

Izberite storitev Montaža Bojlerja
 Količina

Dodaj
Prekliči

Polja označena z * so obvezna.

Slika 9: Dodajanje storitev

Dodajanje storitve fakturi poteka v več korakih. Sistem iz šifrantskih tabel v podatkovni bazi napolni izbiro količine, ter vse storitve, katere se hranijo v podatkovni bazi. Cena na enoto predstavlja ceno za izbrano storitev ene enote. Količina predstavlja število porabljenih enot za

izvedbo storitve. Po zaključku dodajanja storitve fakturi, sistem v pomnilnik storitev doda storitev z vnešenimi podatki.

Storitve se hranijo v pomnilniku dokler se ne zaključi proces vnosa fakture. Takšen pristop sistemu omogoča obremenitev pomnilnika namesto obremenitve podatkovne baze ter samega beleženja v podatkovni bazi. Prikaz dodanih storitev na uporabniškem vmesniku izvaja sistem z xsl transformacijo, katere postopek delovanja je prikazan v prejšnjih korakih. Ko se transformacija izvede, sistem v kontrolo z imenom Literal vnese html katerega nam vrne transformacija. Celotna stran se nato osveži kot html stran. Proces odstranjevanja storitev iz seznama, poteka iz strani uporabnika preko javascript funkcije. Xsl transformacija omogoča pri kreiranju html kode tudi poizvedbe javascript. Tako se v xsl transformaciji, vsaki fakturi določi ključ, kateri se iz klienta posreduje strežniku. Strežnik nato odstrani storitev iz seznama, ter ponovno proži transformacijo. Stran se nato osveži.

2.4.4.4 Pregled faktur

Pregled faktur je poleg vnosa fakture glavna funkcionalnost informacijskega sistema evidenca faktur. Da bi stranki omogočili enostavno poizvedovanje po fakturah, pregledovanje faktur omogoča selekcijo po šestih opcijskih kriterijih: leto, vrsta fakture, naziv stranke, datum, po vrednosti fakture, po statusu. Sistem omogoča tudi validiranje vseh vnešenih kriterijev, ter tako obvesti uporabnika o človeških napakah.

Prikaz pregleda faktur:

Pregled

Kriteriji pregleda:

<input checked="" type="checkbox"/> Leto:	2013 ▼	
<input checked="" type="checkbox"/> Vrsta fakture:	Račun ▼	
<input type="checkbox"/> Naziv stranke:		
<input type="checkbox"/> Datum:	Datum od: <input type="text"/>	Datum do: <input type="text"/>
<input type="checkbox"/> Vrednost:	Vrednost od: <input type="text"/>	Vrednost do: <input type="text"/>
<input type="checkbox"/> Status:	2-Vnešena ▼	

Skupaj zapisov: 5

Šifra fakture	Vrsta	Naziv	Naslov	Davčna številka	Datum	Vrednost brez ddv	Vrednost ddv	Skupaj	Davek(%)	Status	
1	Račun	Test d.o.o.	UL 25 maja 20 a 6258 Prestranek	12345678	17.02.2013	1,00	,20	1,20	20	Oddano	PDF
2	Predračun	Miha Potočnik	Cesta v mestni log 10, 1000 Ljubljana	11111111	30.04.2013	4.036,00	807,20	4.843,20	20	Oddano	PDF
3	Račun	Miha Potočnik	Cesta v mestni log 10, 1000 Ljubljana	11111111	30.04.2013	1,00	,20	1,20	20	Oddano	PDF
4	Račun	Miha Potočnik	Cesta v mestni log 10, 1000 Ljubljana	11111111	20.05.2013	400,00	80,00	480,00	20	Oddano	PDF
5	Račun	Test d.o.o.	UL 25 maja 20 a 6258 Prestranek	12345678	20.05.2013	130,00	26,00	156,00	20	Oddano	PDF

Slika 10: Pregled faktur

Pregled faktur omogoča iskanje izdanih faktur na uporabniku učinkovit in preprost način. Za vsako fakturo omogoča uporabniku izvoz v pdf formatu. Sestavlja ga pet glavnih funkcionalnosti: opcijski kriteriji, dinamična poizvedba, prikaz podatkov, beleženje pregleda, izvoz v pdf format.

- Opcijski kriteriji: Opcijski kriteriji uporabniku omogočajo iskanje faktur preko točno določenega kriterija. Pred vsakim izmed kriterijev je stikalo, katerega uporabnik vključi v primeru upoštevanja kriterija in izključi v primeru ne upoštevanja kriterija. Vsak izmed upoštevanih kriterijev se ob iskanju validira, ter ob napaki opozori uporabnika. S tem sistem zagotavlja pravilnost poizvedbe, ki se nato izvrši nad podatkovno bazo.
- Dinamična poizvedba: Veliko sistemov pri visokih obremenitvah ter pri veliki množici podatkov imajo preformačne težave. S procesom optimizacije poizvedb lahko preformance sistema nekoliko izboljšamo. Sistem Evidenca faktur za obvladovanje preformančnih težav uporablja dinamično poizvedbo ter predpripravljene podatke. Dinamična poizvedba se generira ob vsakem iskanju, tako zajame le izbrane kriterije. V primeru da poizvedba preverja ali je kriterij izpolnjen ali ne, se čas izvajanja poizvedbe lahko zelo podaljša. Sistem uporablja tudi v naprej pripravljene podatke v indeksiranem vpogledu. To pomeni da se v podatkovni bazi v vpogledu vw_Fakture nahajajo vsi potrebni podatki za selekcijo ter iskanje po kriterijih. Seleksijska polja imajo še indeks, kar dodatno pohitri poizvedbo.
- Prikaz podatkov: Sistem pri pregledu faktur, podatke prikazuje preko html kode, katera se generira s pomočjo xsl transformacije. Iz podatkovne baze se podatki preberejo v kontrolo podatkovni set katerega sestavljajo podatkovne tabele. Iz podatkovnega seta sistem pridobi xml dokument, kateri se nato s pomočjo transformacije xsl pretvori v html kodo. Html koda se nato prikaže v kontroli z imenom Literal.
- Beleženje pregleda: Zaradi varnosti ter nadzora nad uporabniki ter zaščite podatkov, se vsak pregled beleži. Sistem izvede beleženje pred iskanjem, tako da preko metode v podatkovni bazi zabeleži uporabniško ime, kriterije, datum, kodo strani, ime računalnika, iz katerega se je pregled izvedel, v tabelo namenjeno nadzoru pregledov. V primeru da ima uporabnik dostop do strani za pregledovanje faktur, lahko pride do zlorab ali kakršnih koli napak. Sistemski administrator uporabnikov lahko tako lažje preverja zlorabe ali napake, ki se pojavijo v sistemu.
- Izvoz v pdf: Sistem omogoča izdajanje faktur na tak način, da podatke o fakturi pretvori v pdf format ter ponudi uporabniku. Preko transformacije xsl se vsaki fakturi doda gumb PDF, kateri nato preko javascript funkcije iz klienta poda zahtevo strežniku za izvoz določene fakture. Strežnik nato iz podatkovne baze preko shranjene metode, katera kot vhodni parameter dobi enolični identifikator fakture, pridobi xml dokument kateri vsebuje glavo fakture ter storitve. Xml dokument se nato preko transformacije xsl transformira v html kodo. S pomočjo plačljive knjižnice Winnovative se nato html koda pretvori v pdf format. Orodje winnovative kot vhod vzame html kodo, kot izhod pa vrne pdf v byte formatu. Nato se koda v byte formatu preko spletnega vmesnika zapiše na uporabnikovo stran, v obliki pdf kot priponka. Uporabnik lahko nato fakturo v pdf formatu shrani, zavrne ali odpre, ter nato v nadaljevanju tudi natisne.

V primeru, da se izvoz določene fakture izvede prvič, sistem fakture označi kot izdano. V tem primeru se izračuna tudi rok plačila oziroma valuta, ki je sedem dni po izvozu fakture. Predloga fakture temelji na uradni predlogi obrazca za samostojne podjetnike. Vsebuje prostor za žig, podatke o stranki, podatke o storitvah ter ceni, podatke o roku dela ter roku plačila, ter podatke o izdajatelju fakture kateri so: Kraj bivališča, transakcijski račun, banka transakcijskega računa.

Prikaz izdane fakture:

Zoom in (Ctrl+Plus)

Izdajatelj:

Prejemnik:

Miha Potočnik
Cesta v mestni log 10, 1000 Ljubljana

Račun št.: 7

Prestranek, dne 25.05.2013
Datum opravljanja storitve od: 01.05.2013
Datum opravljanja storitve do: 14.05.2013
Valuta: 01.06.2013

Plaćilo na trr: 10100-0035811716
pri BANKA KOPER

Enota	Količina	Naziv	Cena na enoto
URA	20	Montaža Bojlerja	25.00
URA	48	Montaža kotla	25.00
KOS	17	Montiranje radiatorja	20.00
KOS	17	Napeljava cevi za radiator	30.00

Skupaj [brez ddv]

2550.00

DDV[20]

510.00

Skupaj za plačilo

3060.00

Vse cene so v valuti euro.

Slika 11: Faktura v formatu PDF

2.5 Mobilna aplikacija

2.5.1 Splošno

Mobilne aplikacije so računalniški programi, prilagojeni za delovanje na mobilnih napravah kot so pametni telefoni, tablični računalniki, itd. Uporabnikom omogočajo preprostejšo uporabo določenih storitev. Mobilne aplikacije so se pojavile ob prihodu pametnih telefonov, tako imenovanih smartphones, ki se zelo hitro razvijajo. Poleg osnovnih funkcionalnosti, kot so: klicanje, pošiljanje kratkih sporočil, nam pametni telefoni omogočajo dostop do spleta ter izvajanje mobilnih aplikacij. Z uporabo mobilnih operacijskih sistemov kot so android, windows phone ter ios, so pametni telefoni postali preprosto mini prenosni računalnik.

Vse mobilne aplikacije temeljijo oz. so razvite za določen operacijski sistem. Poznamo več vrst operacijskih sistemov, najbolj uporabljeni pa so Android od podjetja Google, ios od podjetja Apple ter windows phone od podjetja Microsoft. Mobilne aplikacije, ki je razvita za Android operacijski sistem, ni mogoče namestiti na drug operacijski sistem. Mobilne aplikacije so danes že zelo razvite in se še nenehno razvijajo. Podpirajo že skoraj vsako storitev kot je elektronsko bančništvo, prebiranje novic, pregledovanje podatkov, servisiranje avtomobilov, itd. Tako lahko po uporabi mobilne aplikacije ločimo v nekaj podzvrsti. Poznamo poslovne mobilne aplikacije, ki podpirajo poslovne procese kot so mobilno bančništvo, pregledovanje podatkov v avtomobilskem računalniku, dostop do določenih podatkov. Poleg poslovnih mobilnih aplikacij poznamo tudi mobilne aplikacije s pomočjo katerih se lahko učimo. Poznamo aplikacije za prebiranje knjig, aplikacije za nakupe knjig preko spleta, razne tečaje tujih jezikov. Najbolj popularne mobilne aplikacije pa so vsekakor računalniške igrice. Obstaja jih zelo veliko, tako za najmlajše kot tudi za zahtevne uporabnike.

Mobilne aplikacije lahko ločimo še v dve kategoriji in sicer brezplačne in plačljive. Lastnik mobilne aplikacije se lahko odloči, da bo tržil svoj produkt oz. mobilno aplikacijo, ali pa da jo bo uporabnikom ponudil zastonj. Vse mobilne aplikacije se dobi na enem mestu, tako brezplačne kot plačljive. Za operacijski sistem Android lahko uporabnik dostopa do tako imenovanega Android market-a, na katerem si lahko sam izbira mobilne aplikacije. Za operacijski sistem IOS pa je na voljo App store.

2.5.2 Opreacijski sistem Android

Operacijski sistem Android je odprtokodni operacijski sistem v lasti podjetja Google. Temelji na jedru Linux in je zasnovan za uporabo zaslona na dotik na pametnih telefonih. Do leta 2012 je bilo razvitih in ponujenih preko 700.000 mobilnih aplikacij. Danes ta številka presega 800.000, zato je Android s približno 51% deležem najbolj priljubljen mobilni operacijski sistem.

Razvijanje mobilnih aplikacij za operacijski sistem Android poteka v jeziku xml ter jeziku java. Jezik xml je uporabljen za definicijo uporabniškega grafičnega vmesnika, sama funkcionalnost mobilne aplikacije pa je razvita v javi. Eno izmed najbolj priljubljenih orodji za razvijanje Android mobilnih aplikacij je Eclipse, ki omogoča tudi simuliranje aplikacije v simulatorju ali na mobilnem pametnem telefonu preko USB priključka. Vsako aplikacijo orodje samodejno zapakira v paket s končnico .apk, katero enostavno prenesemo na mobilni telefon ter namestimo.

2.5.3 Mobilna aplikacija »evidenca faktur«

Mobilna aplikacija v sklopu informacijskega sistema »evidenca faktur« je razvita v orodju Eclipse za odprtokodni operacijski sistem Android. Namenjena je podpori poslovnih procesov kateri so pregledovanja ter zaključevanje fakture. Mobilna aplikacija je kompatibilna z verzijami operacijskega sistema od Android 2.2.3 do Android 4.2. Mobilna aplikacija je preprosta za uporabo, ob enem pa uporabniku omogoče vpogled v izdane nepotrijejene oz. nazaključene fakture ter zaključevanje oz. potrjevanje le-teh.

2.5.3.1 Povezava na podatkovno bazo

Iz varnostnih razlogov se mobilne aplikacije za operacijski sistem Android ne povezujejo neposredno na podatkovno bazo. V večini primerih se zato uporablja tako imenovane spletne servise. Spletni servisi so metode za izmenjavo podatkov med dvema zaprtima sistemoma preko protokola http. Pošiljajo se paketi v obliki SOAP. Podatke si lahko izmenjujejo v različnih formatih. Poleg osnovnih formatov kateri so števila, tekst, bit, lahko spletni servisi izmenjujejo tudi formate xml ter json (javascript object). Poznamo več vrst spletnih servisov:

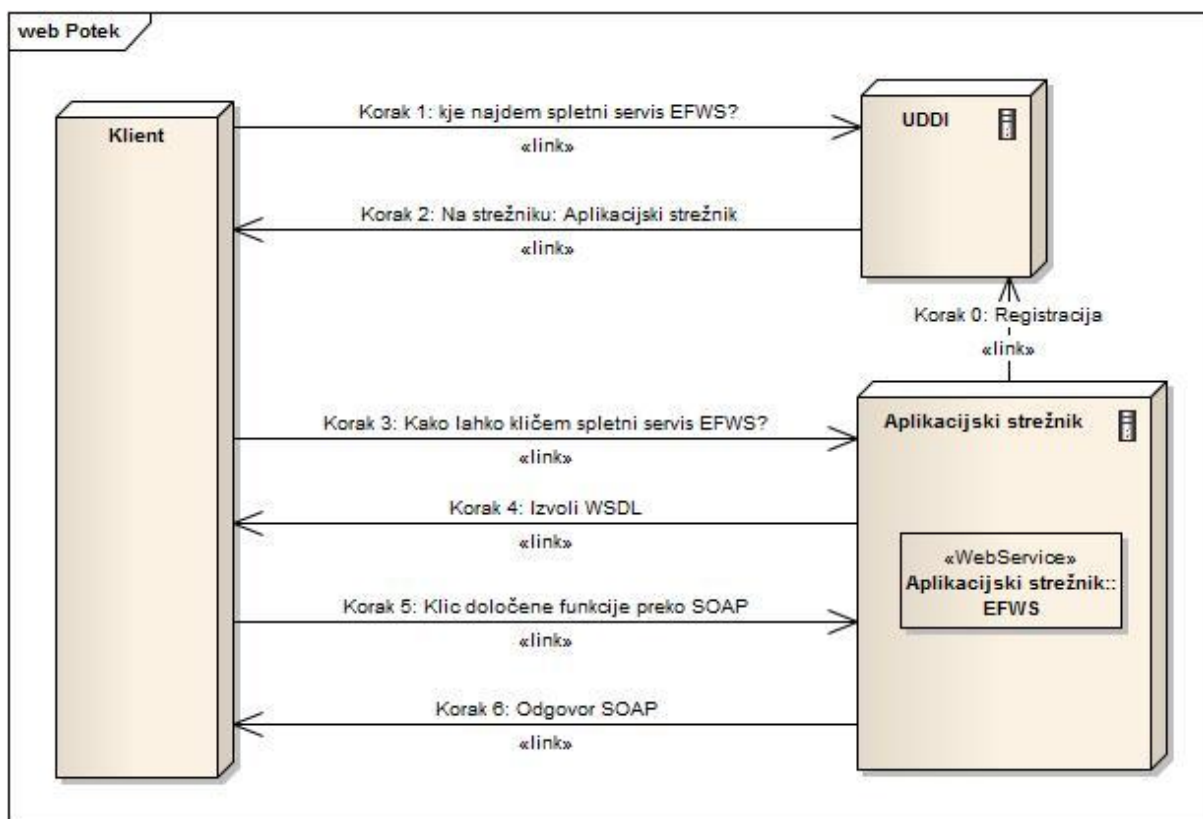
- G2G je elektronska izmenjava podatkov in/ali informacijskih sistemov, med vladnimi agencijami, službami in organizacijami. Cilj G2G je podpora e-vladne pobude za izboljšanje komunikacije, dostop in izmenjavo podatkov.
- B2B je elektronska izmenjava podatkov s ciljem uporabe skupnih spletnih strani vladnih organizacij oz. korporacij. Za uporabo centralnega sistema.
- B2C je elektronska izmenjava podatkov med korporacijami in uporabniki. Tipičen primer takšnih spletnih servisov ima google, npr. za določanje razdalje od ene do druge točke, google maps, ...

Ker mobilna aplikacija za podporo procesom informacijskega sistema »evidenca faktur«, potrebuje podatke iz podatkovne baze Microsoft sql server 2008, je bilo potrebno kreirati spletni servis za izmenjavo podatkov med mobilno aplikacijo in podatkovno bazo. Gre za spletni servis B2B, kateri je kreiran v orodju Microsoft visual studio 2010 in vsebuje štiri metaode:

- Metoda HelloWorld: Metoda HelloWorld uporabniku ob klicu spletnega servisa in njegove metode vrne tekstovni podatkovni niz »Hello word«. Namen metode HelloWorld je preprosto preverjanje dosegljivosti spletnega servisa.
- Metoda GetList: Metoda GetList je namenjena pridobivanju podatkov iz podatkovne baze. Pridobiva se seznam faktur, katere so bile izdane vendar še niso potrjene. Metoda ne potrebuje parametrov. Seznam faktur katere še niso potrjene je vrnjen v obliki json(javascript object), zaradi najlažjega razčlenjevanja podatkov v mobilni aplikaciji.
- Metoda GetDetails: Metoda GetDetails je namenjena pridobivanju podrobnosti o fakturi. Kot parameter klica sprejme enolični identifikator fakture, za katero nato vrne podatke v obliki json.
- Metoda Update: Metoda Update je namenjena potrjevanju faktur, tako pridobi kot vhodni parameter enolični identifikator fakture, kateri na podatkovni bazi spremeni status iz izdana v potrjena(zaključena).

Metode za pridobivanje podatkov vračajo odgovor v obliki json. Rezultat iz podatkovne baze shranijo v json objekt, ter ga nato deserilizirajo v tekst, katerega posredujejo v odgovoru. Json je javascript object notation, namenjen izmenjavi podatkov, ravno tako kot xml. Je človeško berljiv in se je uveljavil kot alternativa xml obliki.

Spletni servis je naložen na dostopnem serverju, v povezavi s podatkovno bazo zaradi manipuliranja s podatki. Na server se je preko IISM (*internet information services manager*) naložil paket, potreben za delovanje spletnega servisa. Paket vsebuje .dll datoteko, ki vsebuje delovanje spletnih servisov, .asmx datoteko, ki uporabniku prikaže WSDL (kakšne metode obstajajo ter kakšni parametri) ter konfiguracijsko datoteko, v kateri se konfigurira delovanje ter povezovanje na podatkovno bazo.



Slika 12: Prikaz delovanja spletnega servisa

UDDI: UDDI oziroma angleški izraz *Universal Description Discovery Integration* je standard za opisovanje, objavljane in iskanje spletnih storitev. Temelji na tehnologiji xml. Komunikacija poteka preko SOAP, CORBA ali Java protokolov. UDDI je namenjen temu, da uporabniku vrne željen naslov o spletni storitvi ter podatke o njej. Podpirajo ga največje računalniške korporacije kot so Microsoft, Dell, SAP, IBM, Intel, Oracle, Sun, HP, itd. Komunikacija med klientom in UDDI poteka preko svetovnega spleta.

WSDL: WSDL oziroma angleški izraz *Web Services Description Language* je standard za opisovanje spletnega servisa. WSDL temelji na XML tehnologiji, z njim se spletni servis predstavi v UDDI, ter daje navodila klientu, na kakšen način je možen klic spletnega servisa oziroma storitve.

SOAP: SOAP oziroma angleški izraz *Simple Object Access Protocol* je protokol za dostop do spletnih storitev oziroma spletnih servisov. Temelji na tehnologiji XML, vsebuje pa glavo in telo. V glavi SOAP se nahajajo specifični podatki o spletnem servisu, kot so na primer avtentikacijski podatki. V telesu SOAP pa so podatki o klicu spletnega servisa. To so ime metode, parametri ter napake.

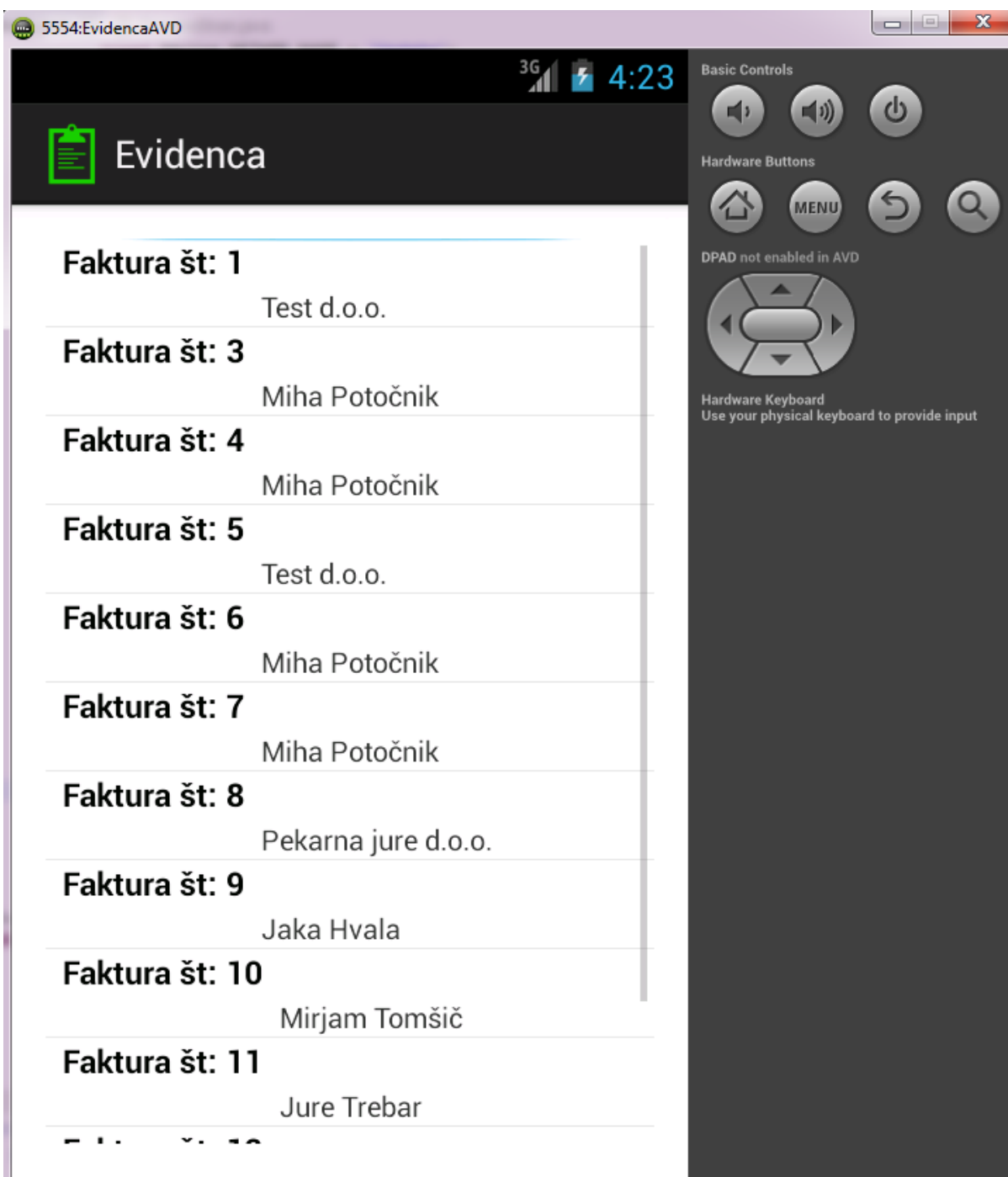
Klic spletnega servisa od začetka spletnega servisa do odgovora klientu, sestavlja sedem korakov:

- Korak 0: Vsaka spletna storitev se najprej registrira v UDDI. Posreduje svoj WSDL ter metapodatke. UDDI nato shrani podatke.
- Korak 1: Klient poišče podatke o spletnem servisu v UDDI. Preko svetovnega spleta klient pošlje zahtevo o podatkih za določeno spletno storitev.
- Korak 2: UDDI preko svetovnega spleta odgovori klientu ter mu posreduje podatke o spletnem servisu. Najpomembnejši podatek je, kje se spletni servis nahaja.
- Korak 3: Klient pošlje zahtevo na aplikacijski strežnik, ki je pridobil za zahtevo opisa spletne storitve(WSDL).
- Korak 4: Aplikacijski strežnik klientu posreduje opis spletne storitve (WSDL), iz katerega klient razbere katere metode so na razpolago v spletnem servisu in na kakšen način je mogoče poizvedovanje.
- Korak 5: Klient sestavi zahtevo (SOAP), ki vsebuje avtentikacijo na spletni servis, željeno metodo ter parametre. Zahtevo pošlje spletni storitvi.
- Korak 6: Spletna storitev po končanem procesiranju zahteve sestavi odgovor (SOAP) ter ga pošlje klientu. Tako se interakcije zaključijo, klient pa pridobi željene podatke oz izvede željeno akcijo.

2.5.3.2 Delovanje mobilne aplikacije

Mobilna aplikacija za podporo poslovnim procesom pregledovanja ter potrjevanja izdanih faktur je implementirana v okolju Eclipse za operacijski sistem Android. Aplikacija je kompatibilna z Android verzijami, večjimi od 2.2.3. Uporabniški vmesnik je razvit v jeziku xml, funkcionalnost pa v programskem jeziku java. Mobilna aplikacija vsebuje dva uporabniška vmesnika. Prvi vmesnik omogoča uporabniku pregled vseh nezaključenih faktur. Omenjeni vmesnik uporabniku prikaže drsni seznam s številko fakture in nazivom stranke. Po izbiri točno določene fakture aplikacija prikaže uporabniku podrobnosti o fakturi kot so številka fakture, datum vnosa fakture, podatki o stranki, podatki o vrednosti fakture, stopnjo davka, vrednost davka ter gumb za zaključitev fakture. Aplikacija omogoča tudi potrjevanje fakture, na način pritiska gumba zaključiti.

Prikaz uporabniškega vmesnika mobilne aplikacije:



Slika 13: Prikaz seznama v mobilni aplikaciji

Ob zagonu mobilne aplikacije se uporabniku prikaže drsni seznam faktur, ki spadajo med račune in so nepotrjene oziroma nezaključene. Aplikacija podatke preko spletne storitve prebere iz podatkovne baze ter jih prikaže na zaslonu. Aplikacija najprej preko protokola SOAP pošlje zahtevo na spletno storitev EFWS za metodo GetList, katera iz podatkovne baze prebere seznam, sestavljen iz številke fakture in naziva pripadajoče stranke, ga pretvori v JSON objekt in ga kot tekst vrne aplikaciji kot odgovor. Aplikacija nato pridobi SOAP odgovor, v katerem je JSON tekst, ga prebere in ga serilizira v JSON objekt. Za tem korakom se aplikacija sprehodi čez vse elemente v seznamu in jih preko unikatnega adapterja posreduje

v drsni seznam. Vsak element v drsnem seznamu je svoj objekt in je posebj oblikovan v posebni xml datoteki. Zaradi tega je potreben unikaten adapter, ki vhodne parametre, v našem primeru številko fakture ter naziv stranke, shrani v objekt in ga doda v drsni seznam, kateri se nato prikaže na zaslonu. Ko se polnjenje seznama zaključi, se v primeru izbire oziroma klika fakture, na drsni seznam doda poslušalec. Ob izbiri oziroma kliku na določeno fakture poslušalec na drsnem seznamu omogoča izvajanje željene akcije. Ob izbiri določene fakture aplikacija novi aktivnosti dodeli izbrani enolični identifikator fakture ter jo zažene.

Enolični identifikator fakture se shrani v tako imenovano inherit spremenljivko, katera nam omogoča prenos določenih vrednosti med različnimi aktivnostimi. Na pregledu se pojavi novo okno, ki uporabniku prikaže podrobnosti o izbrani fakturi ter omogoči zaključevanje fakture.



Slika 14: Prikaz podrobnosti ter zaključevanje fakture

Nov uporabniški vmesnik se odpre po izbiri določene fakture iz drsnega seznama. Kot vhodni parameter potrebuje enolični identifikator izbrane fakture. Po izbiri fakture aplikacija preko SOAP protokola pošlje zahtevo na spletno storitev EFWS za metodo GetDetails. Metoda GetDetails sprejme kot vhodni parameter enolični identifikator fakture, za katero nato pridobi podatke kot so datum izdaje, naziv ter naslov stranke, davčno številko, vrednost fakture, davčno stopnjo fakture, vrednost davka na dodano vrednost. Metoda nato te podatke shrani v JSON objekt ter ga kot tekst pošlje v SOAP odgovoru nazaj aplikaciji. Ko aplikacija prejme odgovor od spletne storitve, tekst serilizira v JSON objekt, ga prebere ter prikaže na uporabniškem vmesniku mobilne aplikacije, kot prikazuje slika številka 14.

Gumb z imenom zaključi uporabniku omogoča zaključitev izbrane fakture. Zaključevanje poteka preko spletne storitve tako, da aplikacija preko SOAP protokola pošlje zahtevo na EFWS spletno storitev, za metodo imenovano Update. V SOAP paket vključi tudi vhodni parameter metode, enolični identifikator fakture. Ko spletna storitev pridobi zahtevo za Update z vhodnim parametrom, posodobi status na podatkovni bazi za določeno fakturo, na faktura zaključena. V primeru, da se operacija uspešno izvede, spletna storitev vrne SOAP odgovor z vsebino »OK«. V nasprotnem primeru pa z vsebino napake. Tako mobilna aplikacija pridobi informacijo o uspešnosti izvršitve ukaza. V primeru, da je aplikacija pridobila odgovor o uspešni izvršitvi ukaza, prikaže na novo osvežen drsni seznam ne zaključenih faktur.

3 MOŽNOSTI RAZŠIRITEV SISTEMA

Informacijski sistem Evidenca faktur trenutno omogoča izvajane funkcionalnosti sistema samo za enega točno določenega uporabnika. Sistem pa je zasnovan za delovanje za več strank, kar omogoča tudi možnost nadgradnje. V primeru trženja ter objave spletne aplikacije na spletu so predvidene naslednje nadgradnje:

- Varnost: Varnost sistema ob prijavi uporabnika bi morala biti povečana. Z uporabo digitalnih potrdil bi bilo za varno avtentikacijo poskrbljeno. Ker pa je izdaja digitalnih potrdil draga, bi uporabil kriptiranje s pomočjo dodatne kode strežnika.

Zagotoviti bi bilo potrebno tudi varnost podatkov ter podatkovne baze. Podatkovna baza bi morala imeti tudi rezervni strežnik, na katerega bi se izdelovala kopija s pomočjo transakcijske replikacije ali celo podatkovne baze v oblaku.

Potrebno bi bilo zagotoviti tudi varnost mobilne aplikacije ter dostop do spletnih storitev. Z uporabo digitalnih potrdil bi lahko varno dostopali do spletne storitve, z dodatnim geslom pa bi lahko zaščitili mobilno aplikacijo. Podoben primer uporabe zaščite se uporablja za mobilno spletno bančništvo.

- Izgled aplikacije: vmesnik bi bilo potrebno nadgraditi v bolj pregledni ter estetski vmesnik, z uporabo tehnologije jquery ter css. Za uporabnika je pri izboru estetika pomembna.
- Razširitev na poljubno mnogo strank: Trenutno sistem deluje samo za točno določeno stranko. Sistem je zasnovan tako, da bi v primeru nadgradnje deloval za več strank hkrati. Tako bi lahko imela vsaka stranka posebej uporabnike, ki bi lahko operirali z njenimi fakturami.

4 ZAKLJUČEK

Skozi diplomsko delo sem opredelil uporabo ter znanja, pridobljena na tehniškem šolskem centru Nova Gorica, na Fakulteti za računalništvo in informatiko v Ljubljani ter v nadaljnjem delu. Uporabil sem znanja načrtovanja, izvedbe ter testiranja informacijskih sistemov ter jih utrdil skozi praktično izvedbo spletne aplikacije s podatkovno bazo.

Cilj diplomske naloge je bil učinkovito uporabljanje virov za implementacijo informacijskega sistema za podporo evidence faktur določenega samostojnega podjetnika, z možnostjo nadaljne nadgraditve v primeru trženja sistema. Uporabil sem vsa do sedaj pridobljena znanja .net programiranja v orodju Microsoft visual studio 2010, programiranja T-SQL v orodju Microsoft SQL server managment studio, znanja spletnih servisov in java mobilnega programiranja.

LITERATURA IN VIRI

[1] (2013) ASP,

Dostopno na: <http://www.asp.net/>

[2] (2013) Winnovative,

Dostopno na: <http://www.winnovative-software.com/Html-To-Pdf-Converter.aspx>

[3] (2013) W3Schools,

Dostopno na: <http://www.w3schools.com>

[4] (2013) MD5,

Dostopno na: <http://www.iwebtool.com/md5>

[5] (2013) Microsoft VisualStudio 2010,

Dostopno na: <http://www.microsoft.com/visualstudio/eng/products/visual-studio-2010-express>

[6] (2013) T-SQL,

Dostopno na: <http://social.msdn.microsoft.com/Forums/en-US/transactsql/threads>

[7] (2013) Microsoft MSDN,

Dostopno na: <http://msdn.microsoft.com/en-US/>

[8] (2013) Enterprise architect,

Dostopno na: <http://www.sparxsystems.com.au/>

[9] (2013) Lucidchart,

Dostopno na: <https://www.lucidchart.com/>

[10] (2013) Spletne storitve,

Dostopno na: http://en.wikipedia.org/wiki/Web_service

[11] (2013) Internet information services,

Dostopno na: [http://msdn.microsoft.com/en-us/library/aa716212\(v=vs.60\).aspx](http://msdn.microsoft.com/en-us/library/aa716212(v=vs.60).aspx)

[12] (2013) Android,

Dostopno na: <http://developer.android.com/>



Št. naloge: 00420 / 2013
Datum: 5.4.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MARTIN BUBNIČ**

Naslov: **SISTEM ZA POTRJEVANJE RAČUNOV**
THE SYSTEM FOR AUTHORISATION OF INVOICES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V večjih podjetjih prejemajo veliko računov, ki morajo čez zahtevne postopke potrjevanja. Fizično prenašanje računov po podjetju je zato lahko težavno, še zlasti v primerih, ko ima podjetje lokacije razpršene po celotni državi. Zasnуйте sistem za zajem in potrjevanje računov. Sistem naj sestavljata spletna aplikacij in mobilna aplikacija, ki obe pooblaščenim posameznikom omogoča potrjevanje računov glede na hierarhično shemo potrjevanja. Spletno aplikacijo razvijte z orodjem Microsoft Visual studio 2010 in z uporabo podatkovne baze MS SQL Server 2008. Mobilna aplikacija naj bo razvita za platformo Android.

Mentor:

doc. dr. Rok Rupnik



Dekan:

prof. dr. Nikolaj Žimic